

1. Reverse Word Game

Reverse is a simple word game where the objective is to find as many *Reverse* word pairs as possible. Two words make a *Reverse* pair if one word is reverse of the other (for example **top** and **pot**).

Given a collection of words (no more than 1000 words).
Your task is to find out the number of reverse pairs among them.

Implement a function that receives an array of strings and returns the total number of reverse pairs on table.

```
//C++
int reversePairsCount (vector<string> words) {
    // Your code here
}
```

```
//Java
int reversePairsCount (List<String> words) {
    // Your code here
}
```

```
//C#
int ReversePairsCount (List<string> words) {
    // Your code here
}
```

//Example:

Input	Output
top aaa read stop aaa table pots stop pot	3

2. Laser Strike

In a Laser Strike game it is critically important that the game field size is a prime number. So help developers to find the smallest prime number for the game field size that is not used yet.

Implement a function that receives an array of integers and returns the smallest prime number that is not included in the array.

There will be no more than 1000 integers in the array.

```
//C++
int smallestPrime (int* numbers, int size) {
    // Your code here
}
```

```
//Java
int smallestPrime (int[] numbers) {
    // Your code here
}
```

```
//C#
int SmallestPrime (int[] numbers) {
    // Your code here
}
```

Example:

Input	Output
2, 3, 5, 4, 3 ,10, 13, 12	7

3. Three Bishops

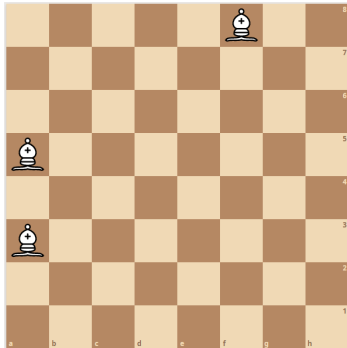
There are three bishops on a chessboard.

In each move a player can take one of his bishops and move it according to chess rules.

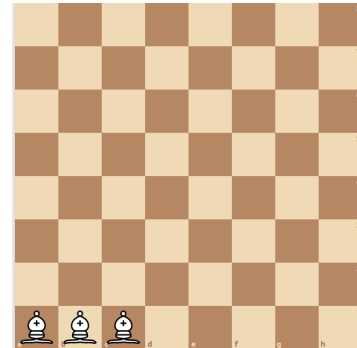
In this game several bishops can occupy the same cell.

Determine how many moves the player needs to place all his bishops in the same cell.

In some cases it can be impossible.



3 Moves needed



Impossible

Implement a function that receives coordinates of 3 bishops (6 integer numbers in interval [1,8]) and returns the minimum number of moves needed to place all the bishops in the same cell, if it is impossible the function should return -1.

//C++

```
int minMoves (int x1, int y1, int x2, int y2, int x3, int y3) {  
    // Your code here  
}
```

//Java

```
int minMoves (int x1, int y1, int x2, int y2, int x3, int y3) {  
    // Your code here  
}
```

//C#

```
int MinMoves (int x1, int y1, int x2, int y2, int x3, int y3) {  
    // Your code here  
}
```

Input	Output
1 3 1 5 6 8	3

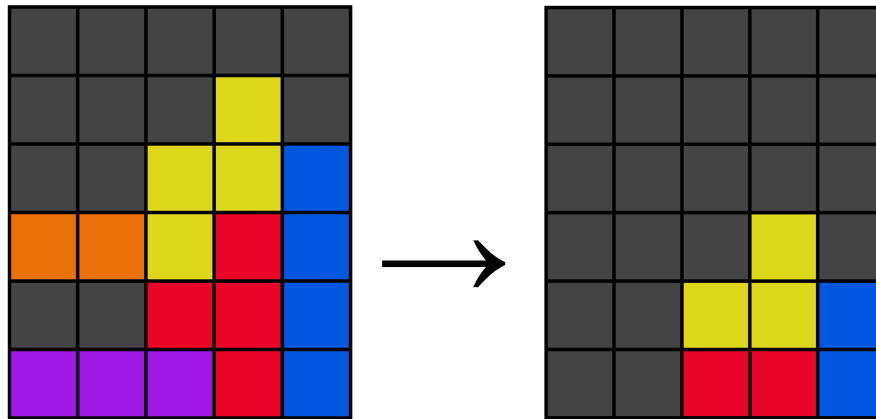
4. Moving Down

In a Tetris game if the player fills the horizontal lines, all completed lines disappear and the lines above move down. Help developers to implement a function that gets the tetris game board as a two-dimensional array of integers and modifies it according to Tetris rules.

Tetris board is represented by rectangular matrix of integers with $0 < height < 100$ and $0 < width < 100$.

The cell of the matrix is 0, if it is empty, and 1 if it is occupied by block.

For example, the following board should be transformed the following way, because 2 lines should disappear and the lines above should move down.



The initial board will be represented as

```
0 0 0 0 0
0 0 0 1 0
0 0 1 1 1
1 1 1 1 1
0 0 1 1 1
1 1 1 1 1
```

After removing completed lines and moving down lines above we should get this matrix.

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 1 0
0 0 1 1 1
0 0 1 1 1
```

```
//C++  
void moveDown (int** board, int height, int width) {  
    // Your code here  
}
```

```
//Java  
void moveDown (int[][] board, int height, int width) {  
    // Your code here  
}
```

```
//C#  
void MoveDown (int[,] board, int height, int width) {  
    // Your code here  
}
```